ARTICLE IN PRESS

Cairo University

Egyptian Informatics Journal

REVIEW

Recommendation systems: Principles, methods and evaluation

F.O. Isinkaye [a,*], Y.O. Folajimi [b], B.A. Ojokoh [c]

[a] *Department of Mathematical Science, Ekiti State University, Ado Ekiti, Nigeria*
[b] *Department of Computer Science, University of Ibadan, Ibadan, Nigeria*
[c] *Department of Computer Science, Federal University of Technology, Akure, Nigeria*

Abstract   On the Internet, where the number of choices is overwhelming, there is need to filter, prioritize and efficiently deliver relevant information in order to alleviate the problem of information overload, which has created a potential problem to many Internet users. Recommender systems solve this problem by searching through large volume of dynamically generated information to provide users with personalized content and services. This paper explores the different characteristics and potentials of different prediction techniques in recommendation systems in order to serve as a compass for research and practice in the field of recommendation systems.

© 2015 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University.

Contents

* Corresponding author.
E-mail address: sadeisinkaye@gmail.com (F.O. Isinkaye).

1. Introduction

The explosive growth in the amount of available digital information and the number of visitors to the Internet have created a potential challenge of information overload which hinders timely access to items of interest on the Internet. Information retrieval systems, such as Google, DevilFinder and Altavista have partially solved this problem but prioritization and personalization (where a system maps available content to user's interests and preferences) of information were absent. This has increased the demand for recommender systems more than ever before. Recommender systems are information filtering systems that deal with the problem of information overload [1] by filtering vital information fragment out of large amount of dynamically generated information according to user's preferences, interest, or observed behavior about item [2]. Recommender system has the ability to predict whether a particular user would prefer an item or not based on the user's profile.

Recommender systems are beneficial to both service providers and users [3]. They reduce transaction costs of finding and selecting items in an online shopping environment [4]. Recommendation systems have also proved to improve decision making process and quality [5]. In e-commerce setting, recommender systems enhance revenues, for the fact that they are effective means of selling more products [3]. In scientific libraries, recommender systems support users by allowing them to move beyond catalog searches. Therefore, the need to use efficient and accurate recommendation techniques within a system that will provide relevant and dependable recommendations for users cannot be over-emphasized.

2. Related work

Recommender system is defined as a decision making strategy for users under complex information environments [6]. Also,

recommender system was defined from the perspective of E-commerce as a tool that helps users search through records of knowledge which is related to users' interest and preference [7]. Recommender system was defined as a means of assisting and augmenting the social process of using recommendations of others to make choices when there is no sufficient personal knowledge or experience of the alternatives [8]. Recommender systems handle the problem of information overload that users normally encounter by providing them with personalized, exclusive content and service recommendations. Recently, various approaches for building recommendation systems have been developed, which can utilize either collaborative filtering, content-based filtering or hybrid filtering [9–11]. Collaborative filtering technique is the most mature and the most commonly implemented. Collaborative filtering recommends items by identifying other users with similar taste; it uses their opinion to recommend items to the active user. Collaborative recommender systems have been implemented in different application areas. GroupLens is a news-based architecture which employed collaborative methods in assisting users to locate articles from massive news database [12]. Ringo is an online social information filtering system that uses collaborative filtering to build users profile based on their ratings on music albums [10]. Amazon uses topic diversification algorithms to improve its recommendation [13]. The system uses collaborative filtering method to overcome scalability issue by generating a table of similar items offline through the use of item-to-item matrix. The system then recommends other products which are similar online according to the users' purchase history. On the other hand, content-based techniques match content resources to user characteristics. Content-based filtering techniques normally base their predictions on user's information, and they ignore contributions from other users as with the case of collaborative techniques [14,15]. Fab relies heavily on the ratings of different users in order to create a training set and it is an example of content-based recommender system. Some

other systems that use content-based filtering to help users find information on the Internet include Letizia [16]. The system makes use of a user interface that assists users in browsing the Internet; it is able to track the browsing pattern of a user to predict the pages that they may be interested in. Pazzani et al. [17] designed an intelligent agent that attempts to predict which web pages will interest a user by using naive Bayesian classifier. The agent allows a user to provide training instances by rating different pages as either hot or cold. Jennings and Higuchi [18] describe a neural network that models the interests of a user in a Usenet news environment.

Despite the success of these two filtering techniques, several limitations have been identified. Some of the problems associated with content-based filtering techniques are limited content analysis, overspecialization and sparsity of data [12]. Also, collaborative approaches exhibit cold-start, sparsity and scalability problems. These problems usually reduce the quality of recommendations. In order to mitigate some of the problems identified, Hybrid filtering, which combines two or more filtering techniques in different ways in order to increase the accuracy and performance of recommender systems has been proposed [19,20]. These techniques combine two or more filtering approaches in order to harness their strengths while leveling out their corresponding weaknesses [21]. They can be classified based on their operations into weighted hybrid, mixed hybrid, switching hybrid, feature-combination hybrid, cascade hybrid, feature-augmented hybrid and meta-level hybrid [22]. Collaborative filtering and content-based filtering approaches are widely used today by implementing content-based and collaborative techniques differently and the results of their prediction later combined or adding the characteristics of content-based to collaborative filtering and vice versa. Finally, a general unified model which incorporates both content-based and collaborative filtering properties could be developed [12]. The problem of sparsity of data and cold-start was addressed by combining the ratings, features and demographic information about items in a cascade hybrid recommendation technique in [23]. In Ziegler et al. [24], a hybrid collaborative filtering approach was proposed to exploit bulk taxonomic information designed for exacting product classification to address the data sparsity problem of CF recommendations, based on the generation of profiles via inference of super-topic score and topic diversification. A hybrid recommendation technique is also proposed in Ghazantar and Pragel-Benett [23], and this uses the content-based profile of individual user to find similar users which are used to make predictions. In Sarwar et al. [25], collaborative filtering was combined with an information filtering agent. Here, the authors proposed a framework for integrating the content-based filtering agents and collaborative filtering. A hybrid recommender algorithm is employed by many applications as a result of new user problem of content-based filtering techniques and average user problem of collaborative filtering [26]. A simple and straightforward method for combining content-based and collaborative filtering was proposed by Cunningham et al. [27]. A music recommendation system which combined tagging information, play counts and social relations was proposed in Konstas et al. [28]. In order to determine the number of neighbors that can be automatically connected on a social platform, Lee and Brusilovsky [29] embedded social information into collaborative filtering algorithm. A Bayesian mixed-effects model that integrates user ratings, user and item features in a single unified framework was proposed by Condiff et al. [30].

3. Phases of recommendation process

3.1. Information collection phase

This collects relevant information of users to generate a user profile or model for the prediction tasks including user's attribute, behaviors or content of the resources the user accesses. A recommendation agent cannot function accurately until the user profile/model has been well constructed. The system needs to know as much as possible from the user in order to provide reasonable recommendation right from the onset. Recommender systems rely on different types of input such as the most convenient high quality explicit feedback, which includes explicit input by users regarding their interest in item or implicit feedback by inferring user preferences indirectly through observing user behavior [31]. Hybrid feedback can also be obtained through the combination of both explicit and implicit feedback. In E-learning platform, a user profile is a collection of personal information associated with a specific user. This information includes cognitive skills, intellectual abilities, learning styles, interest, preferences and interaction with the system. The user profile is normally used to retrieve the needed information to build up a model of the user. Thus, a user profile describes a simple user model. The success of any recommendation system depends largely on its ability to represent user's current interests. Accurate models are indispensable for obtaining relevant and accurate recommendations from any prediction techniques.

3.1.1. Explicit feedback

The system normally prompts the user through the system interface to provide ratings for items in order to construct and improve his model. The accuracy of recommendation depends on the quantity of ratings provided by the user. The only shortcoming of this method is, it requires effort from the users and also, users are not always ready to supply enough information. Despite the fact that explicit feedback requires more effort from user, it is still seen as providing more reliable data, since it does not involve extracting preferences from actions, and it also provides transparency into the recommendation process that results in a slightly higher perceived recommendation quality and more confidence in the recommendations [32].
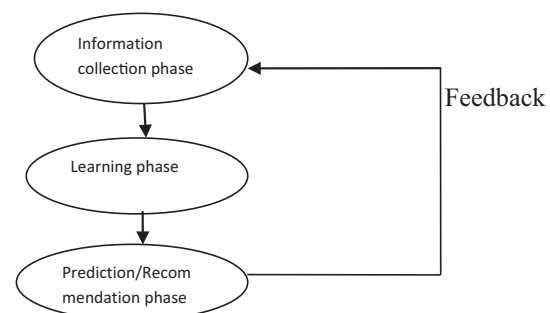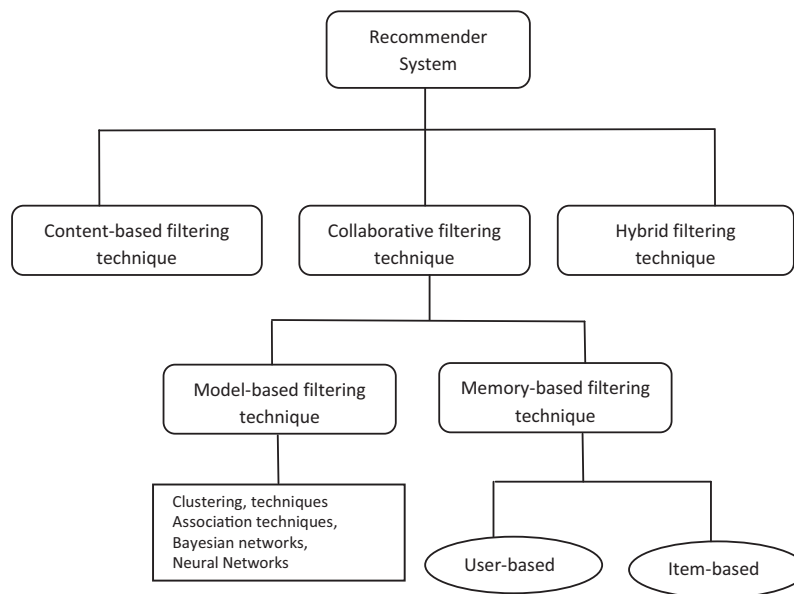

Figure 1   Recommendation phases.

Figure 2   Recommendation techniques.

3.1.2. Implicit feedback

The system automatically infers the user's preferences by monitoring the different actions of users such as the history of purchases, navigation history, and time spent on some web pages, links followed by the user, content of e-mail and button clicks among others. Implicit feedback reduces the burden on users by inferring their user's preferences from their behavior with the system. The method though does not require effort from the user, but it is less accurate. Also, it has also been argued that implicit preference data might in actuality be more objective, as there is no bias arising from users responding in a socially desirable way [32] and there are no self-image issues or any need for maintaining an image for others [33].

3.1.3. Hybrid feedback

The strengths of both implicit and explicit feedback can be combined in a hybrid system in order to minimize their weaknesses and get a best performing system. This can be achieved by using an implicit data as a check on explicit rating or allowing user to give explicit feedback only when he chooses to express explicit interest.

3.2. Learning phase

It applies a learning algorithm to filter and exploit the user's features from the feedback gathered in information collection phase.

### 3.3. Prediction/recommendation phase

It recommends or predicts what kind of items the user may prefer. This can be made either directly based on the dataset collected in information collection phase which could be memory based or model based or through the system's observed activities of the user. Fig. 1 highlights the recommendation phases.

4. Recommendation filtering techniques

The use of efficient and accurate recommendation techniques is very important for a system that will provide good and useful recommendation to its individual users. This explains the importance of understanding the features and potentials of different recommendation techniques. Fig. 2 shows the anatomy of different recommendation filtering techniques.

4.1. Content-based filtering

Content-based technique is a domain-dependent algorithm and it emphasizes more on the analysis of the attributes of items in order to generate predictions. When documents such as web pages, publications and news are to be recommended, content-based filtering technique is the most successful. In content-based filtering technique, recommendation is made based on the user profiles using features extracted from the content of the items the user has evaluated in the past [34,35]. Items that are mostly related to the positively rated items are recommended to the user. CBF uses different types of models to find similarity between documents in order to generate meaningful recommendations. It could use Vector Space Model such as Term Frequency Inverse Document Frequency (TF/IDF) or Probabilistic models such as Naïve Bayes Classifier [36], Decision Trees [37] or Neural Networks [38] to model the relationship between different documents within a corpus. These techniques make recommendations by learning the underlying model with either statistical analysis or machine learning techniques. Content-based filtering technique does not need the profile of other users since they do not influence recommendation. Also, if the user profile changes, CBF technique still has the potential to adjust its recommendations within a very short period of time. The major disadvantage of this technique is the need to have an in-depth knowledge and description of the features of the items in the profile.
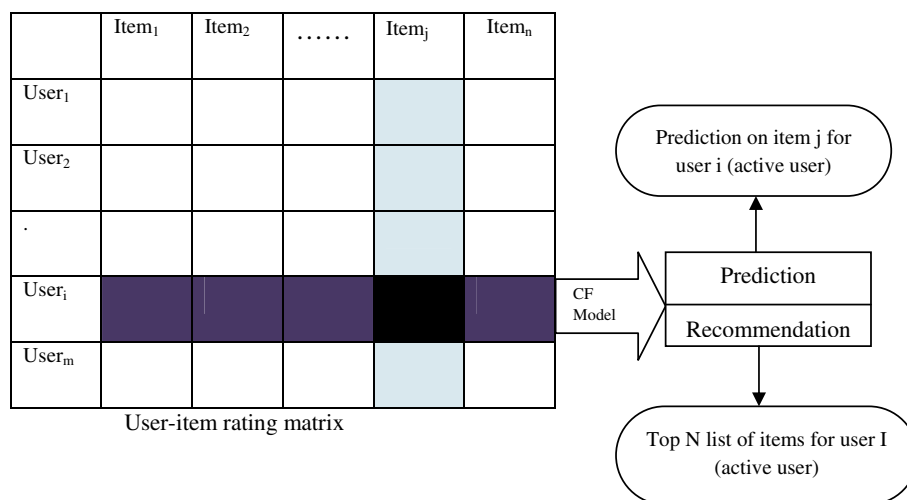
Figure 3   Collaborative filtering process.

4.1.1. Pros and Cons of content-based filtering techniques

CB filtering techniques overcome the challenges of CF. They have the ability to recommend new items even if there are no ratings provided by users. So even if the database does not contain user preferences, recommendation accuracy is not affected. Also, if the user preferences change, it has the capacity to adjust its recommendations in a short span of time. They can manage situations where different users do not share the same items, but only identical items according to their intrinsic features. Users can get recommendations without sharing their profile, and this ensures privacy [39]. CBF technique can also provide explanations on how recommendations are generated to users. However, the techniques suffer from various problems as discussed in the literature [12]. Content based filtering techniques are dependent on items' metadata. That is, they require rich description of items and very well organized user profile before recommendation can be made to users. This is called limited content analysis. So, the effectiveness of CBF depends on the availability of descriptive data. Content over-specialization [40] is another serious problem of CBF technique. Users are restricted to getting recommendations similar to items already defined in their profiles.

4.1.2. Examples of content-based filtering systems

News Dude [41] is a personal news system that utilizes synthesized speech to read news stories to users. TF-IDF model is used to describe news stories in order to determine the short-term recommendations which is then compared with the Cosine Similarity Measure and finally supplied to a learning algorithm (NN). CiteSeer is an automatic citation indexing that uses various heuristics and machine learning algorithms to process documents. Today, CiteSeer is among the largest and widely used research paper repository on the web.

LIBRA [42] is a content-based book recommendation system that uses information about book gathered from the Web. It implements a Naïve Bayes classifier on the information extracted from the web to learn a user profile to produce a ranked list of titles based on training examples supplied by an individual user. The system is able to provide explanation on any recommendations made to users by listing the features

that contribute to the highest ratings and hence allowing the users to have total confidence on the recommendations provided to users by the system.

4.2. Collaborative filtering

Collaborative filtering is a domain-independent prediction technique for content that cannot easily and adequately be described by metadata such as movies and music. Collaborative filtering technique works by building a database (user-item matrix) of preferences for items by users. It then matches users with relevant interest and preferences by calculating similarities between their profiles to make recommendations [43]. Such users build a group called neighborhood. An user gets recommendations to those items that he has not rated before but that were already positively rated by users in his neighborhood. Recommendations that are produced by CF can be of either prediction or recommendation. Prediction is a numerical value, Rij, expressing the predicted score of item j for the user i, while Recommendation is a list of top N items that the user will like the most as shown in Fig. 3. The technique of collaborative filtering can be divided into two categories: memory-based and model-based [35,44].

4.2.1. Memory based techniques

The items that were already rated by the user before play a relevant role in searching for a neighbor that shares appreciation with him [45,46]. Once a neighbor of a user is found, different algorithms can be used to combine the preferences of neighbors to generate recommendations. Due to the effectiveness of these techniques, they have achieved widespread success in real life applications. Memory-based CF can be achieved in two ways through user-based and item-based techniques. User based collaborative filtering technique calculates similarity between users by comparing their ratings on the same item, and it then computes the predicted rating for an item by the active user as a weighted average of the ratings of the item by users similar to the active user where weights are the similarities of these users with the target item. Item-based filtering techniques compute predictions using the similarity between

items and not the similarity between users. It builds a model of item similarities by retrieving all items rated by an active user from the user-item matrix, it determines how similar the retrieved items are to the target item, then it selects the k most similar items and their corresponding similarities are also determined. Prediction is made by taking a weighted average of the active users rating on the similar items k. Several types of similarity measures are used to compute similarity between item/user. The two most popular similarity measures are correlation-based and cosine-based. Pearson correlation coefficient is used to measure the extent to which two variables linearly relate with each other and is defined as [47,48]

$$s(a,u) = \frac{\sum_{i=1}^{n}(r_{a,i} - \overline{r_a})(r_{u,i} - \overline{r_u})}{\sqrt{\sum_{i=1}^{n}(r_{a,i} - \overline{r_a})^2}\sqrt{\sum_{i=1}^{n}(r_{u,i} - \overline{r_u})^2}} \qquad (1)$$

From the above equation, $s(a,u)$ denotes the similarity between two users a and u, $r_{a,i}$ is the rating given to item i by user a, $\overline{r_a}$ is the mean rating given by user a while n is the total number of items in the user-item space. Also, prediction for an item is made from the weighted combination of the selected neighbors' ratings, which is computed as the weighted deviation from the neighbors' mean. The general prediction formula is

$$p(a,i) = \overline{r_a} + \frac{\sum_{i=1}^{n}(r_{u,i} - \overline{r_u}) \times s(a,u)}{\sum_{i=1}^{n}s(a,u)} \qquad (2)$$

Cosine similarity is different from Pearson-based measure in that it is a vector-space model which is based on linear algebra rather that statistical approach. It measures the similarity between two n-dimensional vectors based on the angle between them. Cosine-based measure is widely used in the fields of information retrieval and texts mining to compare two text documents, in this case, documents are represented as vectors of terms. The similarity between two items u and v can be defined as [12,43,48] follows:

$$s(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| * |\vec{v}|} = \frac{\sum_i r_{u,i} r_{v,i}}{\sqrt{\sum_i r_{u,i}^2} \times \sqrt{\sum_i r_{v,i}^2}} \qquad (3)$$

Similarity measure is also referred to as similarity metric, and they are methods used to calculate the scores that express how similar users or items are to each other. These scores can then be used as the foundation of user- or item-based recommendation generation. Depending on the context of use, similarity metrics can also be referred to as correlation metrics or distance metrics [12].

4.2.2. Model-based techniques

This technique employs the previous ratings to learn a model in order to improve the performance of Collaborative filtering Technique. The model building process can be done using machine learning or data mining techniques. These techniques can quickly recommend a set of items for the fact that they use pre-computed model and they have proved to produce recommendation results that are similar to neighborhood-based recommender techniques. Examples of these techniques include Dimensionality Reduction technique such as Singular Value Decomposition (SVD), Matrix Completion Technique, Latent Semantic methods, and Regression and Clustering. Model-based techniques analyze the user-item matrix to identify relations between items; they use these relations to compare the list of top-N recommendations. Model based techniques resolve the sparsity problems associated with recommendation systems.

The use of learning algorithms has also changed the manner of recommendations from recommending what to consume by users to recommending when to actually consume a product. It is therefore very important to examine other learning algorithms used in model-based recommender systems:

Association rule: Association rules mining algorithms [49] extract rules that predict the occurrence of an item based on the presence of other items in a transaction. For instance, given a set of transactions, where each transaction is a set of items, an association rule applies the form A → B, where A and B are two sets of items [50]. Association rules can form a very compact representation of preference data that may improve efficiency of storage as well as performance. Also, the effectiveness of association rule for uncovering patterns and driving personalized marketing decisions has been known for sometimes [2]. However, there is a clear relation between this method and the goal of a Recommendation System but they have not become mainstream.

Clustering: Clustering techniques have been applied in different domains such as, pattern recognition, image processing, statistical data analysis and knowledge discovery [51]. Clustering algorithm tries to partition a set of data into a set of sub-clusters in order to discover meaningful groups that exist within them [52]. Once clusters have been formed, the opinions of other users in a cluster can be averaged and used to make recommendations for individual users. A good clustering method will produce high quality clusters in which the intra-cluster similarity is high, while the inter-cluster similarity is low. In some clustering approaches, a user can have partial participation in different clusters, and recommendations are then based on the average across the clusters of participation which is weighted by degree of participation [53]. K-means and Self-Organizing Map (SOM) are the most commonly used among the different clustering methods. K-means takes an input parameter, and then partitions a set of n items into K clusters [54]. The Self-Organizing Map (SOM) is a method for an unsupervised learning, based on artificial neurons clustering technique [55]. Clustering techniques can be used to reduce the candidate set in collaborative-based algorithms.

Decision tree: Decision tree is based on the methodology of tree graphs which is constructed by analyzing a set of training examples for which the class labels are known. They are then applied to classify previously unseen examples. If trained on very high quality data, they have the ability to make very accurate predictions [56]. Decision trees are more interpretable than other classifier such as Support Vector machine (SVM) and Neural Networks because they combine simple questions about data in an understandable manner. Decision trees are also flexible in handling items with mixture of real-valued and categorical features as well as items that have some specific missing features.

Artificial Neural network: ANN is a structure of many connected neurons (nodes) which are arranged in layers in systematic ways. The connections between neurons have weights associated with them depending on the amount of

influence one neuron has on another. There are some advantages in using neural networks in some special problem situations. For example, due to the fact that it contains many neurons and also assigned weight to each connection, an artificial neural network is quite robust with respect to noisy and erroneous data sets [57]. ANN has the ability of estimating nonlinear functions and capturing complex relationships in data sets also, they can be efficient and even operate if part of the network fails. The major disadvantage is that it is hard to come up with the ideal network topology for a given problem and once the topology is decided this will act as a lower bound for the classification error.

Link analysis: Link Analysis is the process of building up networks of interconnected objects in order to explore pattern and trends [58]. It has presented great potentials in improving the accomplishment of web search. Link analysis consists of PageRank and HITS algorithms. Most link analysis algorithms handle a web page as a single node in the web graph [59].

Regression: Regression analysis is used when two or more variables are thought to be systematically connected by a linear relationship. It is a powerful and diversity process for analyzing associative relationships between dependent variable and one or more independent variables. Uses of regression contain curve fitting, prediction, and testing systematic hypotheses about relationships between variables. The curve can be useful to identify a trend within dataset, whether it is linear, parabolic, or of some other forms.

Bayesian Classifiers: They are probabilistic framework for solving classification problems which is based on the definition of conditional probability and Bayes theorem. Bayesian classifiers [36] consider each attribute and class label as random variables. Given a record of N features $(A_1, A_2, \ldots, A_N)$, the goal of the classifier is to predict class C_k by finding the value of C_k that maximizes the posterior probability of the class given the data $P(C_k|A_1, A_2, \ldots, A_N)$ by applying Bayes' theorem, $P(C_k|A_1, A_2, \ldots, A_N) \propto P(A_1, A_2, \ldots, A_N|C_k)P(C_k)$. The most commonly used Bayesian classifier is known as the Naive Bayes Classifier. In order to estimate the conditional probability, $P(A_1, A_2, \ldots, A_N|C_k)$, a Naive Bayes Classifier assumes the probabilistic independence of the attributes that is, the presence or absence of a particular attribute is unrelated to the presence or absence of any other. This assumption leads to $P(A_1, A_2, \ldots, A_N|C_k) = P(A_1|C_k)P(A_2|C_k)\ldots P(A_N|C_k)$. The main benefits of Naive Bayes classifiers are that they are robust to isolated noise points and irrelevant attributes, and they handle missing values by ignoring the instance during probability estimate calculations. However, the independence assumption may not hold for some attributes as they might be correlated. In this case, the usual approach is to use Bayesian Networks. Bayesian classifiers may prove practical for environments in which knowledge of user preferences changes slowly with respect to the time needed to build the model but are not suitable for environments in which users preference models must be updated rapidly or frequently. It is also successful in model-based recommendation systems because it is often used to derive a model for content-based recommendation systems.

Matrix completion techniques: The essence of matrix completion technique is to predict the unknown values within the user-item matrices. Correlation based K-nearest neighbor is one of the major techniques employed in collaborative filtering recommendation systems [60]. They depend largely on the historical rating data of users on items. Most of the time, the rating matrix is always very big and sparse due to the fact that users do not rate most of the items represented within the matrix [61]. This problem always leads to the inability of the system to give reliable and accurate recommendations to users. Different variations of low rank models have been used in practice for matrix completion especially toward application in collaborative filtering [62]. Formally, the task of matrix completion technique is to estimate the entries of a matrix, $M \in \mathbb{R}^{m \times n}$, when a subset, $\Omega \subset \{(i,j) : 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n\}$ of the new entries is observed, a particular set of low rank matrices, $\widehat{M} = UV^T$, where $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{m \times k}$ and $k \ll \min(m, n)$. The most widely used algorithm in practice for recovering M from partially observed matrix using low rank assumption is Alternating Least Square (ALS) minimization which involves optimizing over U and V in an alternating manner to minimize the square error over observed entries while keeping other factors fixed. Candes and Recht [63] proposed the use of matrix completion technique in the Netflix problem as a practical example for the utilization of the technique. Keshavan et al. [64] used SVD technique in an OptSpace algorithm to deal with matrix completion problem. The result of their experiment showed that SVD is able provide a reliable initial estimate for spanning subspace which can be further refined by gradient descent on a Grassmannian manifold. Model based techniques solve sparsity problem. The major drawback of the techniques is that the model building process is computationally expensive and the capacity of memory usage is highly intensive. Also, they do not alleviate the cold-start problem.

4.2.3. Pros and Cons of collaborative filtering techniques

Collaborative Filtering has some major advantages over CBF in that it can perform in domains where there is not much content associated with items and where content is difficult for a computer system to analyze (such as opinions and ideal). Also, CF technique has the ability to provide serendipitous recommendations, which means that it can recommend items that are relevant to the user even without the content being in the user's profile [65]. Despite the success of CF techniques, their widespread use has revealed some potential problems such as follows.

4.2.3.1. Cold-start problem.
This refers to a situation where a recommender does not have adequate information about a user or an item in order to make relevant predictions [66]. This is one of the major problems that reduce the performance of recommendation system. The profile of such new user or item will be empty since he has not rated any item; hence, his taste is not known to the system.

4.2.3.2. Data sparsity problem.
This is the problem that occurs as a result of lack of enough information, that is, when only a few of the total number of items available in a database are rated by users [34,67]. This always leads to a sparse user-item matrix, inability to locate successful neighbors and finally, the generation of weak recommendations. Also, data sparsity

always leads to coverage problems, which is the percentage of items in the system that recommendations can be made for [68]

4.2.3.3. Scalability. This is another problem associated with recommendation algorithms because computation normally grows linearly with the number of users and items [67]. A recommendation technique that is efficient when the number of dataset is limited may be unable to generate satisfactory number of recommendations when the volume of dataset is increased. Thus, it is crucial to apply recommendation techniques which are capable of scaling up in a successful manner as the number of dataset in a database increases. Methods used for solving scalability problem and speeding up recommendation generation are based on Dimensionality reduction techniques, such as Singular Value Decomposition (SVD) method, which has the ability to produce reliable and efficient recommendations.

4.2.3.4. Synonymy. Synonymy is the tendency of very similar items to have different names or entries. Most recommender systems find it difficult to make distinction between closely related items such as the difference between e.g. baby wear and baby cloth. Collaborative Filtering systems usually find no match between the two terms to be able to compute their similarity. Different methods, such as automatic term expansion, the construction of a thesaurus, and Singular Value Decomposition (SVD), especially Latent Semantic Indexing are capable of solving the synonymy problem. The shortcoming of these methods is that some added terms may have different meanings from what is intended, which sometimes leads to rapid degradation of recommendation performance.

4.2.4. Examples of collaborative systems

Ringo [69] is a user-based CF system which makes recommendations of music albums and artists. In Ringo, when a user initially enters the system, a list of 125 artists is given to the user to rate according to how much he likes listening to them. The list is made up of two different sections. The first session consists of the most often rated artists, and this affords the active user opportunity to rate artists which others have equally rated, so that there is a level of similarities between different users' profiles. The second session is generated upon a random selection of items from the entire user-item matrix, so that all artists and albums are eventually rated at some point in the initial rating phases.

GroupLens [70] is a CF system that is based on client/server architecture; the system recommends Usenet news which is a high volume discussion list service on the Internet. The short lifetime of Netnews, and the underlying sparsity of the rating matrices are the two main challenges addressed by this system. Users and Netnews are clustered based on the existing news groups in the system, and the implicit ratings are computed by measuring the time the users spend reading Netnews.

Amazon.com is an example of e-commerce recommendation engine that uses scalable item-to-item collaborative filtering techniques to recommend online products for different users. The computational algorithm scales independently of the number of users and items [53] within the database. Amazon.com uses an explicit information collection technique to obtain information from users. The interface is made up of the following sections, your browsing history, rate these items, and improve your recommendations and your profile. The system predicts users interest based on the items he/she has rated. The system then compares the users browsing pattern on the
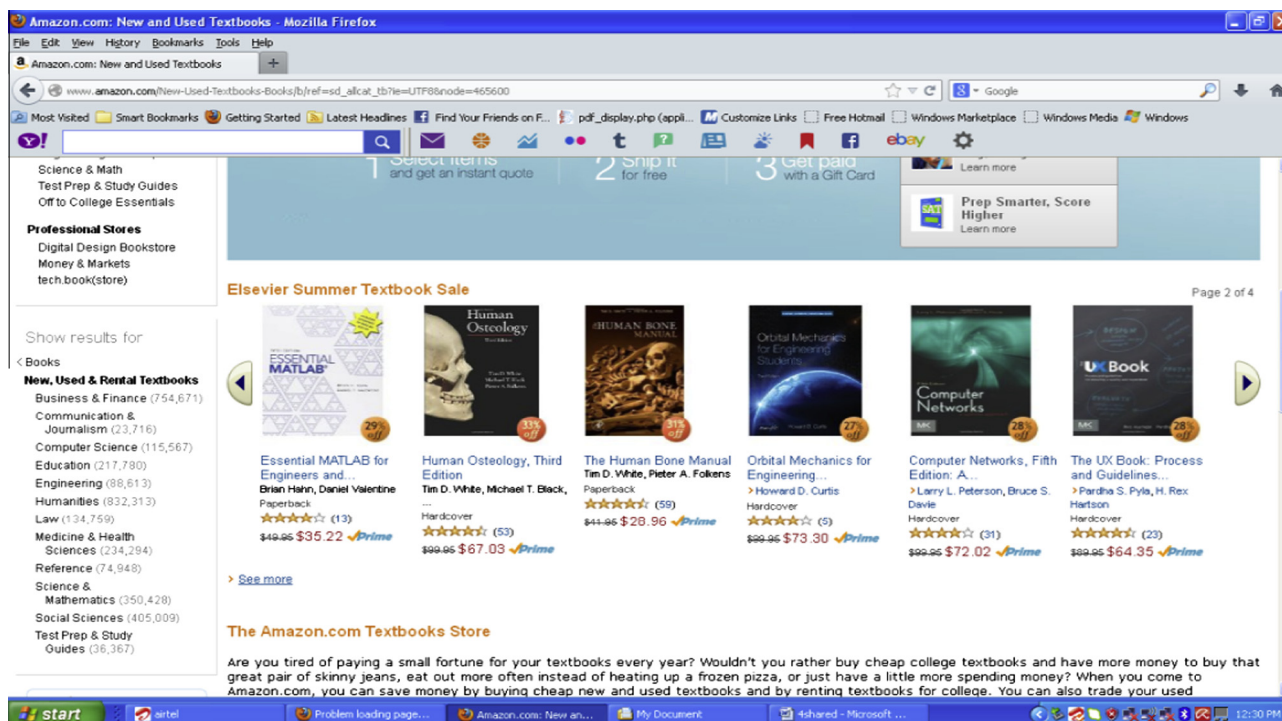

Figure 4   Amason book recommender interface. *Sourced from:* www.amason.com.

system and decides the item of interest to recommend to the user [71]. Amazon.com popularized feature of "people who bought this item also bought these items". Example of Amazon.com item-to-item contextual recommendation interface is shown in Fig. 4.

4.2.5. Trust in collaborative filtering recommendation systems

Trust in RS is defined as the correlation between similar preference toward the items that are commonly rated or liked by two users [72]. Trust improves RS by combining similarity and trust between users. That is, the way neighbors are selected is modified by introducing trust in order to develop new relationship between users so that it can increase connectivity and alleviate the challenges of data sparsity and cold start associated with traditional collaborative filtering techniques. Some of the empirical studies conducted by Ziegler et al. [24] revealed that correlation exists between trust and user similarity when community's trust network is bound to some specific application. Following the studies, it can be deduced that computational trust models can act as appropriate means to supplement or completely replace current collaborative filtering technique [73].

Different trust metrics are used in RS to measure and calculate the value between users in a network. These metrics are of two types, local and global trust metrics. *Local trust metrics* used the subjective opinion of the active user to predict the trustworthiness of other users from the active user perspective. The trust value represents the amount of trust that the active user puts on another user. Based on this technique, different users trust the active user differently and therefore their trust value is different from each other. *Global trust metrics* represents an entire community's opinion regarding the current user; therefore, every user receives only one value that represents her level of trustworthiness in the community. Trust scores in global trust metrics are calculated by the aggregation of all users' opinions as regards the current user. Users' reputation on ebay.com is an example of using global trust in an online shopping website. ebay.com calculates user reputation based on the number of users who left positive, negative, or neutral feedback for the items sold by the current user. When the user does not have a specific opinion regarding another user, she usually relies on these aggregated trust scores. Global trust can be further divided into two parts namely profile level and item-level The profile-level trust refers to the general definition of global trust metrics in which it assigns one trust score to every user.

4.3. Hybrid filtering

Hybrid filtering technique combines different recommendation techniques in order to gain better system optimization to avoid some limitations and problems of pure recommendation systems [74,75]. The idea behind hybrid techniques is that a combination of algorithms will provide more accurate and effective recommendations than a single algorithm as the disadvantages of one algorithm can be overcome by another algorithm [65]. Using multiple recommendation techniques can suppress the weaknesses of an individual technique in a combined model. The combination of approaches can be done in any of the following ways: separate implementation of algorithms and combining the result, utilizing some content-based filtering in collaborative approach, utilizing some collaborative filtering in content-based approach, creating a unified recommendation system that brings together both approaches.

4.3.1. Weighted hybridization

Weighted hybridization combines the results of different recommenders to generate a recommendation list or prediction by integrating the scores from each of the techniques in use by a linear formula. An example of a weighted hybridized recommendation system is P-tango [76]. The system consists of a content-based and collaborative recommender. They are given equal weights at first, but weights are adjusted as predictions are confirmed or otherwise. The benefit of a weighted hybrid is that all the recommender system's strengths are utilized during the recommendation process in a straightforward way.

4.3.2. Switching hybridization

The system swaps to one of the recommendation techniques according to a heuristic reflecting the recommender ability to produce a good rating. The switching hybrid has the ability to avoid problems specific to one method e.g. the new user problem of content-based recommender, by switching to a collaborative recommendation system. The benefit of this strategy is that the system is sensitive to the strengths and weaknesses of its constituent recommenders. The main disadvantage of switching hybrids is that it usually introduces more complexity to recommendation process because the switching criterion, which normally increases the number of parameters to the recommendation system has to be determined [34]. Example of a switching hybrid recommender is the DailyLearner [77] that uses both content-based and collaborative hybrid where a content-based recommendation is employed first before collaborative recommendation in a situation where the content-based system cannot make recommendations with enough evidence.

4.3.3. Cascade hybridization

The cascade hybridization technique applies an iterative refinement process in constructing an order of preference among different items. The recommendations of one technique are refined by another recommendation technique. The first recommendation technique outputs a coarse list of recommendations which is in turn refined by the next recommendation technique. The hybridization technique is very efficient and tolerant to noise due to the coarse-to-finer nature of the iteration. EntreeC [34] is an example of cascade hybridization method that used a cascade knowledge-based and collaborative recommender.

4.3.4. Mixed hybridization

Mixed hybrids combine recommendation results of different recommendation techniques at the same time instead of having just one recommendation per item. Each item has multiple recommendations associated with it from different recommendation techniques. In mixed hybridization, the individual performances do not always affect the general performance of a local region. Example of recommender system in this category that uses the mixed hybridization is the PTV system [78] which recommends a TV viewing schedule for a user by combining recommendations from content-based and

collaborative systems to form a schedule. Profinder [79] and PickAFlick [80] are also examples of mixed hybrid systems.

4.3.5. Feature-combination

The features produced by a specific recommendation technique are fed into another recommendation technique. For example, the rating of similar users which is a feature of collaborative filtering is used in a case-based reasoning recommendation technique as one of the features to determine the similarity between items. Pipper is an example of feature combination technique that used the collaborative filter's ratings in a content-based system as a feature for recommending movies [81]. The benefit of this technique is that, it does not always exclusively rely on the collaborative data.

4.3.6. Feature-augmentation

The technique makes use of the ratings and other information produced by the previous recommender and it also requires additional functionality from the recommender systems. For example, the Libra system [42] makes content-based recommendation of books on data found in Amazon.com by employing a naïve Bayes text classifier. Feature-augmentation hybrids are superior to feature-combination methods in that they add a small number of features to the primary recommender.

4.3.7. Meta-level

The internal model generated by one recommendation technique is used as input for another. The model generated is always richer in information when compared to a single rating. Meta-level [17] hybrids are able to solve the sparsity problem of collaborative filtering techniques by using the entire model learned by the first technique as input for the second technique. Example of meta-level technique is LaboUr [82] which uses instant-based learning to create content-based user profile that is then compared in a collaborative manner.

5. Evaluation metrics for recommendation algorithms

The quality of a recommendation algorithm can be evaluated using different types of measurement which can be accuracy or coverage. The type of metrics used depends on the type of filtering technique. Accuracy is the fraction of correct recommendations out of total possible recommendations while coverage measures the fraction of objects in the search space the system is able to provide recommendations for. Metrics for measuring the accuracy of recommendation filtering systems are divided into statistical and decision support accuracy metrics [83]. The suitability of each metric depends on the features of the dataset and the type of tasks that the recommender system will do [36].

Statistical accuracy metrics evaluate accuracy of a filtering technique by comparing the predicted ratings directly with the actual user rating. Mean Absolute Error (MAE) [84], Root Mean Square Error (RMSE) and Correlation are usually used as statistical accuracy metrics. MAE is the most popular and commonly used; it is a measure of deviation of recommendation from user's specific value. It is computed as follows [76]:

$$MAE = \frac{1}{N}\sum_{u,i}|p_{u,i} - r_{u,i}| \qquad (4)$$

where P_{ui} is the predicted rating for user u on item i, $r_{u,i}$ is the actual rating and N is the total number of ratings on the item set. The lower the MAE, the more accurately the recommendation engine predicts user ratings. Also, the Root Mean Square Error (RMSE) is given by Cotter et al. [85] as

$$RMSE = \sqrt{\frac{1}{n}\sum_{u,i}(p_{u,i} - r_{u,i})^2} \qquad (5)$$

Root Mean Square Error (RMSE) puts more emphasis on larger absolute error and the lower the RMSE is, the better the recommendation accuracy.

Decision support accuracy metrics that are popularly used are Reversal rate, Weighted errors, Receiver Operating Characteristics (ROC) and Precision Recall Curve (PRC), Precision, Recall and *F*-measure. These metrics help users in selecting items that are of very high quality out of the available set of items [86]. The metrics view prediction procedure as a binary operation which distinguishes good items from those items that are not good. ROC curves are very successful when performing comprehensive assessments of the performance of some specific algorithms. Precision is the fraction of recommended items that is actually relevant to the user, while recall can be defined as the fraction of relevant items that are also part of the set of recommended items [87]. They are computed as

$$Precision = \frac{Correctly\ recommended\ items}{Total\ recommended\ items} \qquad (6)$$

$$Recall = \frac{Correctly\ recommended\ items}{Total\ useful\ recommended\ items} \qquad (7)$$

F-measure defined below helps to simplify precision and recall into a single metric. The resulting value makes comparison between algorithms and across data sets very simple and straightforward [83].

$$F\text{-measure} = \frac{2PR}{P + R} \qquad (8)$$

Coverage has to do with the percentage of items and users that a recommender system can provide predictions. Prediction may be practically impossible to make if no users or few users rated an item. Coverage can be reduced by defining small neighborhood sizes for user or items [88].

6. Conclusion

Recommender systems open new opportunities of retrieving personalized information on the Internet. It also helps to alleviate the problem of information overload which is a very common phenomenon with information retrieval systems and enables users to have access to products and services which are not readily available to users on the system. This paper discussed the two traditional recommendation techniques and highlighted their strengths and challenges with diverse kind of hybridization strategies used to improve their performances. Various learning algorithms used in generating recommendation models and evaluation metrics used in measuring the quality and performance of recommendation algorithms were discussed. This knowledge will empower researchers and serve as a road map to improve the state of the art recommendation techniques.

References

[1] Konstan JA, Riedl J. Recommender systems: from algorithms to user experience. User Model User-Adapt Interact 2012;22:101–23.

[2] Pan C, Li W. Research paper recommendation with topic analysis. In Computer Design and Applications IEEE 2010;4, pp. V4-264.

[3] Pu P, Chen L, Hu R. A user-centric evaluation framework for recommender systems. In: Proceedings of the fifth ACM conference on Recommender Systems (RecSys'11), ACM, New York, NY, USA; 2011. p. 57–164.

[4] Hu R, Pu P. Potential acceptance issues of personality-ASED recommender systems. In: Proceedings of ACM conference on recommender systems (RecSys'09), New York City, NY, USA; October 2009. p. 22–5.

[5] Pathak B, Garfinkel R, Gopal R, Venkatesan R, Yin F. Empirical analysis of the impact of recommender systems on sales. J Manage Inform Syst 2010;27(2):159–88.

[6] Rashid AM, Albert I, Cosley D, Lam SK, McNee SM, Konstan JA et al. Getting to know you: learning new user preferences in recommender systems. In: Proceedings of the international conference on intelligent user interfaces; 2002. p. 127–34.

[7] Schafer JB, Konstan J, Riedl J. Recommender system in e-commerce. In: Proceedings of the 1st ACM conference on electronic commerce; 1999. p. 158–66.

[8] Resnick P, Varian HR. Recommender system's. Commun ACM 1997;40(3):56–8. http://dx.doi.org/10.1145/245108.24512.

[9] Acilar AM, Arslan A. A collaborative filtering method based on Artificial Immune Network. Exp Syst Appl 2009;36(4):8324–32.

[10] Chen LS, Hsu FH, Chen MC, Hsu YC. Developing recommender systems with the consideration of product profitability for sellers. Int J Inform Sci 2008;178(4):1032–48.

[11] Jalali M, Mustapha N, Sulaiman M, Mamay A. WEBPUM: a web-based recommendation system to predict user future movement. Exp Syst Applicat 2010;37(9):6201–12.

[12] Adomavicius G, Tuzhilin A. Toward the next generation of recommender system. A survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 2005;17(6):734–49.

[13] Ziegler CN, McNee SM, Konstan JA, Lausen G. Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on World Wide Web; 2005. p. 22–32.

[14] Min SH, Han I. Detection of the customer time-variant pattern for improving recommender system. Exp Syst Applicat 2010;37(4):2911–22.

[15] Celma O, Serra X. FOAFing the Music: bridging the semantic gap in music recommendation. Web Semant: Sci Serv Agents World Wide Web 2008;16(4):250–6.

[16] Lieberman H. Letizia: an agent that assists web browsing. In: Proceedings of the 1995 international joint conference on artificial intelligence. Montreal, Canada; 1995. p. 924–9.

[17] Pazzani MJ. A framework for collaborative, content-based and demographic filtering. Artific Intell Rev 1999;13:393–408, No. 5(6).

[18] Jennings A, Higuchi H. A personal news service based on a user model neural network. IEICE Trans Inform Syst 1992;E75-D(2):198–209.

[19] Murat G, Sule GO. Combination of web page recommender systems. Exp Syst Applicat 2010;37(4):2911–22.

[20] Mobasher B. Recommender systems. Kunstliche Intelligenz. Special Issue on Web Mining, BottcherIT Verlag, Bremen, Germany, vol. 3; 2007. p. 41–3.

[21] Al-Shamri MYH, Bharadway KK. Fuzzy-genetic approach to recommender systems based on a novel hybrid user model. Expert Syst Appl 2008;35(3):1386–99.

[22] Mican D, Tomai N. Association ruled-based recommender system for personalization in adaptive web-based applications. In: Proceedings of the 10th international conference on current trends in web engineering (ICWE'10), Berlin, Springer-Verlag, 2010; p. 85–90.

[23] Ghazantar MA, Pragel-Benett A. A scalable accurate hybrid recommender system. In: the 3rd International conference on knowledge discovery and data mining (WKDD 2010), IEEE Computer Society, Washington, DC, USA. <http://eprint.ecs.so-ton.ac.uk/18430>.

[24] Ziegler CN, Lausen G, Schmidt-Thieme L. Taxonomy-driven computation of product recommendations. In: Proceedings of the 13th international conference on information and knowledge management (CIKM '04), Washington, DC, USA; 2004. p. 406–15.

[25] Sarwar BM, Konstan JA, Herlocker JL, Miller B, Riedl JT. Using filtering agents to improve prediction quality in the grouplens research, collaborative filtering system. In: Proceedings of the ACM conference on computer supported cooperative work. NY (USA): ACM New York; 1998. p. 345–54.

[26] Burke R. Hybrid web recommender systems. In: Brusilovsky P, Kobsa A, Nejdl W, editors. The adaptive web, LNCS 4321. Berlin Heidelberg, Germany: Springer; 2007. p. 377–408. http://dx.doi.org/10.1007/978-3-540-72079-9_12.

[27] Cunningham P, Bergmann R, Schmitt S, Traphoner R, Breen S, Smyth B. "WebSell: Intelligent sales assistants for the World Wide Web. In: Proceedings CBR in ECommerce, Vancouver BC; 2001. p. 104–9.

[28] Konstan I, Stathopoulos V, Jose JM. On social networks and collaborative recommendation. In: The proceedings of the 32nd international ACM conference (SIGIR'09), ACM. New York, NY, USA; 2009. p.195–202.

[29] Lee DH, Brusilovsky P. Social networks and interest similarity: the case of CiteULike. In: Proceedings of the 21st ACM conference on Hypertext and Hypermedia (HT'10). ACM. New York, NY, USA; 2010. p. 151–6.

[30] Condiff MK, Lewis DD, Madigan D, Posse C. Bayesian mixed-effects models for recommender systems. In: Proceedings of ACM SIGIR workshop of recommender systems: algorithm and evaluation; 1999.

[31] Oard DW, Kim J. Implicit feedback for recommender systems. In: Proceedings of 5th DELOS workshop on filtering and collaborative filtering; 1998. p. 31–6.

[32] Buder J, Schwind C. Learning with personalized recommender systems: a psychological view. Comput Human Behav 2012;28(1):207–16.

[33] Gadanho SC, Lhuillier N. Addressing uncertainty in implicit preferences. In: Proceedings of the 2007 ACM conference on Recommender Systems (RecSys '07). ACM, New York, NY, USA; 2007. p. 97–104.

[34] Burke R. Hybrid recommender systems: survey and experiments. User Model User-adapted Interact 2002;12(4):331–70.

[35] Bobadilla J, Ortega F, Hernando A, Gutiérrez A. Recommender systems survey. Knowl-Based Syst 2013;46:109–32.

[36] Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Mach Learn 1997;29(2–3):131–63.

[37] Duda RO, Hart PE, Stork DG. Pattern classification. John Wiley & Sons; 2012.

[38] Bishop CM. Pattern recognition and machine learning, vol. 4, no. 4. Springer, New York; 2006.

[39] Shyong K, Frankowski D, Riedl J. Do you trust your recommendations? An exploration of security and privacy issues in recommender systems. In: Emerging trends in information and communication security. Berlin, Heidelberg: Springer; 2006. p. 14–29.

[40] Zhang T, Vijay SI. Recommender systems using linear classifiers. J Mach Learn Res 2002;2:313–34.

[41] Billsus D, Pazzani MJ. User modeling for adaptive news access. User Model User-adapted Interact 2000;10(2–3):147–80.

[42] Mooney RJ, Roy L. Content-based book recommending using learning for text categorization. In: Proceedings of the fifth ACM conference on digital libraries. ACM; 2000. p. 195–204.

[43] Herlocker JL, Konstan JA, Terveen LG, Riedl JT. Evaluating collaborative filtering recommender systems. ACM Trans Inform Syst 2004;22(1):5–53.

[44] Breese J, Heckerma D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th conference on uncertainty in artificial intelligence (UAI-98); 1998. p. 43–52.

[45] Zhao ZD, Shang MS. User-based collaborative filtering recommendation algorithms on Hadoop. In: Proceedings of 3rd international conference on knowledge discovering and data mining, (WKDD 2010), IEEE Computer Society, Washington DC, USA; 2010. p. 478–81. doi: 10.1109/WKDD.2010.54.

[46] Zhu X, Ye HW, Gong S. A personalized recommendation system combining case-based reasoning and user-based collaborative filtering. In: Control and decision conference (CCDC 2009), Chinese; 2009. p. 4026–28.

[47] Melville P, Mooney-Raymond J, Nagarajan R. Content-boosted collaborative filtering for improved recommendation. In: Proceedings of the eighteenth national conference on artificial intelligence (AAAI), Edmonton, Canada; 2002. p. 187–92.

[48] Jannach D, Zanker M, Felfernig A, Friedrich G. Recommender systems – an introduction. Cambridge University Press; 2010.

[49] Mobasher B, Jin X, Zhou Y. Semantically enhanced collaborative filtering on the web. In: Web mining: from web to semantic web. Berlin Heidelberg: Springer; 2004. p. 57–76.

[50] Yoon HC, Jae KK, Soung HK. A personalized recommender system based on web usage mining and decision tree induction. Expert Syst Appl 2002;23:329–42.

[51] Kuželewska U. Advantages of information granulation in clustering al-gorithms. In: Agents and artificial intelligence. NY: Springer; 2013. p. 131–45.

[52] McSherry D. Explaining the pros and cons of conclusions in CBR. In: Calero PAG, Funk P, editors. Proceedings of the European conference on case-based reasoning (ECCBR-04). Madrid (Spain): Springer; 2004. p. 317–30.

[53] Linden G, Smith B, York J. Amazon.com recommendation: item-to-item collaborative filtering. IEEE Internet Comput 2003;7(1):76–80.

[54] Michael JA, Berry A, Gordon S, Linoff L. Data mining techniques. 2nd ed. Wiley Publishing Inc.; 2004.

[55] Hosseini-Pozveh M, Nemartbakhsh M, Movahhedinia N. A multimedia approach for context-aware recommendation in mobile commerce. Int J Comput Sci Inform Secur 2009;3(1).

[56] Caruana R, Niculescu-Mizil A. An empirical comparison of supervised learning algorithms. In: Cohen W. Moore AW, editors. Machine Learning, Proceedings of the twenty-third international conference, ACM, New York; 2003. p. 161–8.

[57] Larose TD. Discovering knowledge in data. Hoboken (New Jersey): John Wiley; 2005.

[58] Berry MJA, Linoff G. Data mining techniques: for marketing, sales, and customer support. New York: Wiley Computer Publishing; 1997.

[59] Deng C, Xiaofe H, Ji-Rong W, Wei-Ying M. Block-level link analysis. In: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval; 2004. p. 440–7.

[60] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. IEEE Comput 2009;8:30–7.

[61] Bojnordi E, Moradi P. A novel collaborative filtering model based on combination of correlation method with matrix completion technique. In: 16th CSI international symposium on artificial intelligence and signal processing (AISP), IEEE; 2012.

[62] Takács G, István P, Bottyán N, Tikk D. Investigation of various matrix factorization methods for large recommender systems. In: IEEE international conference on data mining workshops. ICDMW'08. IEEE; 2008. p. 553–62.

[63] Candès EJ, Recht B. Exact matrix completion via convex optimization. Found Comput Math 2009;9(6):717–72.

[64] Keshavan RH, Montanari A, Sewoong O. Matrix completion from a few entries. IEEE Trans Inform Theor 2010;56(6):2980–98.

[65] Schafer JB, Frankowski D, Herlocker J, Sen S. Collaborative filtering recommender systems. In: Brusilovsky P, Kobsa A, Nejdl W, editors. The Adaptive Web, LNCS 4321. Berlin Heidelberg (Germany): Springer; 2007. p. 291–324. http://dx.doi.org/10.1007/978-3-540-72079-9_9.

[66] Burke R. Web recommender systems. In: Brusilovsky P, Kobsa A, Nejdl W, editors. The Adaptive Web, LNCS 4321. Berlin Heidelberg (Germany): Springer; 2007. p. 377–408. http://dx.doi.org/10.1007/978-3-540-72079-9_12.

[67] Park DH, Kim HK, Choi IY, Kim JK. A literature review and classification of recommender systems research. Expert Syst Appl 2012;39(11):10059–72.

[68] Su X, Khoshgoftaar TM. A survey of collaborative filtering techniques. Adv Artif Intell 2009;4:19.

[69] Shardanand U, Maes P. Social information filtering: algorithms for automating "word of mouth". In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM Press/Addison-Wesley Publishing Co.; 1995. p. 210–7.

[70] Konstan JA, Miller BN, Maltz D, Herlocker JL, Gordon LR, Riedl J. Applying collaborative filtering to usenet news. Commun ACM 1997;40(3):77–87.

[71] Lee TQ, Young P, Yong-Tae P. A time-based approach to effective recommender systems using implicit feedback. Expert Syst Appl 2008;34(4):3055–62.

[72] Shambour Q, Lu J. A trust-semantic fusion-based recommendation approach for e-business applications. Decis Support Syst 2012;54(1):768–80.

[73] O'Donovan J, Smyth B. Trust in recommender systems. In: Proceedings of the 10th international conference on intelligent user interfaces, ACM; 2005. p. 167–74.

[74] Adomavicius G, Zhang J. Impact of data characteristics on recommender systems performance. ACM Trans Manage Inform Syst 2012;3(1).

[75] Stern DH, Herbrich R, Graepel T. Matchbox: large scale online bayesian recommendations. In: Proceedings of the 18th international conference on World Wide Web. ACM, New York, NY, USA; 2009. p. 111–20.

[76] Claypool M, Gokhale A, Miranda T, Murnikov P, Netes D, Sartin M. Combining content- based and collaborative filters in an online newspaper. In: Proceedings of ACM SIGIR workshop on recommender systems: algorithms and evaluation, Berkeley, California; 1999.

[77] Billsus D, Pazzani MJ. A hybrid user model for news story classification. In: Kay J, editor. In: Proceedings of the seventh international conference on user modeling, Banff, Canada. Springer-Verlag, New York; 1999. p. 99–108.

[78] Smyth B, Cotter P. A personalized TV listings service for the digital TV age. J Knowl-Based Syst 2000;13(2–3):53–9.

[79] Wasfi AM. Collecting user access patterns for building user profiles and collaborative filtering. In: Proceedings of the 1999 international conference on intelligent user, Redondo Beach, CA; 1999. p. 57–64.

[80] Burke R, Hammond K, Young B. The FindMe approach to assisted browsing. IEEE Expert 1997;12(4):32–40.

[81] Basu C, Hirsh H, Cohen W. Recommendation as classification: using social and content-based information in recommendation. In: Proceedings of the 15th national conference on artificial intelligence, Madison, WI; 1998. p. 714–20.

[82] Schwab I, Kobsa A, Koychev I. Learning user interests through positive examples using content analysis and collaborative filtering. Draft from Fraunhofer Institute for Applied Information Technology, Germany; 2001.

[83] Sarwar B, Karypis G, Konstan J, Reidl J. Item-based collabora-tive filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, ACM, Hong Kong, China; 2001. p. 295.

[84] Goldberg K, Roeder T, Gupta D, Perkins C. Eigentaste: a constant time collaborative filtering algorithm. Inform Retrieval J 2001;4(2):133–51.

[85] Cotter P, Smyth B. PTV: Intelligent personalized TV guides. In: Twelfth conference on innovative applications of artificial intel-ligence; 2000. p. 957–64.

[86] Sarwar BM, Konstan JA, Herlocker JL, Miller B, Riedl JT. Using filtering agents to improve prediction quality in the grouplens research, collaborative filtering system. In: Proceedings of the ACM conference on computer supported cooperative work. New York (NY, USA): ACM; 1998. p. 345–54.

[87] Drosou M, Pitoura E. Search result diversification. SIGMOD Rec 2010;39(1):41–7.

[88] Papagelis M, Plexousakis D. Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. Int J Eng Appl Artif Intell 2005;18(4):781–9.